UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

ROBOTIZACIÓN DE TAREAS ADMINISTRATIVAS DE GESTORES DE BASES DE DATOS ORACLE

ROBOTIZATION OF ADMINISTRATIVE TASKS OF ORACLE DATABASE MANAGEMENT SYSTEM

Sergio Castellano Gómez
Tutor: Jack Mario Mingo Postiglioni
Ponente: Iván Cantador Gutiérrez

Julio 2018

ROBOTIZACIÓN DE TAREAS ADMINISTRATIVAS DE GESTORES DE BASES DE DATOS ORACLE

ROBOTIZATION OF ADMINISTRATIVE TASKS OF ORACLE DATABASE MANAGEMENT SYSTEMS

AUTOR: Sergio Castellano Gómez
TUTOR: Jack Mario Mingo Postiglioni
PONENTE: Iván Cantador Gutiérrez

Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2018

Resumen

La tecnología se está haciendo más influyente cada día y está cambiando nuestras vidas constantemente tanto personal como profesionalmente, y este Trabajo de Fin de Grado se ha centrado en cómo conseguir que esta tecnología trabaje para nosotros automatizando tareas que actualmente se hacen por personas. La tecnología utilizada para ejecutar estas tareas se llama Robotic Process Automation (RPA). Y el campo en el que queremos aplicar dicha automatización es el de las Bases de Datos. Siendo más específicos, en Bases de Datos de Oracle. El objetivo principal de este trabajo es la automatización de procesos mediante RPA para ayudar a los administradores de sistemas con ciertas tareas.

Primero, estudiamos los Sistemas de Gestión de Bases de Datos de Oracle y algunas de las tareas administrativas para una gestión correcta. Necesitábamos saber cómo esas tareas son resueltas por personas, para automatizarlas posteriormente. También fue necesario estudiar UiPath, que fue el software de RPA utilizado para la automatización de esos procesos administrativos de Bases de Datos de Oracle. Era necesario entender sus características y las herramientas que nos ofrece.

Después de determinar los casos de uso a automatizar, y cómo son actualmente resueltos por administradores de bases de datos, la automatización se diseñó y desarrolló.

Cuando se consiguió una correcta automatización de los procesos, comparamos el rendimiento ofrecido por el RPA con el de un administrador típico, y los resultados fueron más que satisfactorios. El RPA era dos o tres veces más rápido que el experto ejecutando dichas tareas, y no cometía errores. Además, cualquier usuario puede ejecutar el RPA, sin necesidad de que tenga un conocimiento técnico de Administración de Bases de Datos de Oracle.

Otro objetivo de este trabajo era determinar cuándo y cuánto puede ayudarnos un RPA, y llegamos a las siguientes conclusiones:

- Las personas son mejores en cuanto a la intuición y creatividad. Y en tareas cortas y rápidas, son incluso más rápidas que un RPA.
- Un RPA es mucho más rápido ejecutando procesos repetitivos basados en reglas. Además, no comenten errores en las tareas para los que han sido programados.

A modo de corolario podríamos decir que las personas podemos ahorrarnos una gran cantidad de trabajo gracias a la automatización de procesos.

Palabras clave

RPA, automatización, Inteligencia Artificial, Bases de Datos de Oracle, Sistemas de Gestión de Bases de Datos, UiPath.

Abstract

Technology is getting more and more powerful every day and is changing our lives constantly personally and professionally, and in this Bachelor Thesis we focused on how to make this technology to work for us by automatizing tasks that are currently done by humans. The technology used to execute tasks is called Robotic Process Automation (RPA). And the field where we want to apply such automating technology is on databases, more specifically, in Oracle Databases. The main objective is to have automated processes helping humans with some tasks.

First, we studied Database Management Systems and common administration tasks. We needed to know how humans work on those tasks, to automatize them afterwards. It was also needed to study the RPA software called UiPath, which is the one used for automating those Oracle Database administration processes. It was needed to understand well its features and the possibilities that it could give us.

Once we knew the use cases to be automatized, and how they are currently solved by DBAs, we designed the automation and the development was done.

When those automatized processes were working properly, we tested the performance of the RPA and compared it with a human. And the results were more than satisfactory. The RPA was two to three times faster than the human and it was not making any mistakes. Also, anyone could run the RPA, even someone without a technical knowledge on Oracle Databases.

The goal of this work was to determine when to use and how helpful could a RPA be, and we arrived to the following conclusions:

- Humans are better with intuition and creativity and sometimes faster with short processes.
- RPA is much faster with rule-based processes and much more efficient with large tasks. And they do not make mistakes in what they have been programmed for.

This means that people can save a lot of time with the automation of processes.

Keywords

RPA, automation, Artificial Intelligence, Oracle Database, Database Management Systems, UiPath.

Acknowledgements

In the first place I would like to thank my tutor Mario for making this Bachelor Thesis possible. Without him, it would had gone in a totally different direction.

I want to thank my family, especially my mother, and my girlfriend, for giving me constant support during these studies.

I also want to thank my workmates during my internship in Accenture. I got to know about RPA and many other technologies related to Artificial Intelligence thanks to them.

INDEX OF CONTENTS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introduction

1.1 Motivation

Technology is becoming more powerful and useful every year, and most of the types of technology are aimed to make our lives easier or entertain us. This work is focused on the first one, making our lives easier by automating processes.

Computers are faster, more efficient, and make less mistakes than people. And we should take advantage of that and make them do some of the tough work for us through automation. Robot Process Automation (RPA) is a technology which combines Artificial Intelligence (AI) and automation [1], and it has a huge potential.

In this work we studied the Database Management System (DBMS) of Oracle Databases, and some common tasks performed by a Database Administrator (DBA). Being able to automate such technical work would show one more useful field where RPA could be applied.

RPA is able to execute rule-based processes like a human does [2], and we should use that advantage to let it help us. Surely it will be able to automatize much more complex processes in the future, but for now we will focus on long and repetitive tasks.

1.2 Objectives

The main objective of this work is to prove how RPA works with more efficiency than technicians in certain processes. For that, we will study in depth some tasks performed by DBAs.

We want to test and achieve the automation of processes currently performed by experienced DBAs. And after performing that automation check, to analyze the real advantages of implementing RPA automations in any field. We will to focus on processes that do not require interaction with human, and that are time-consuming [3]. This work does not only want to prove that RPA saves time, but also to achieve specific conclusions about the time that it could save to DBAs who are able to use this in their every-day work.

1.3 Organization of the report

This work consists of the following sections:

- *State of technologies.* Here it is studied Oracle Databases and the processes which could be automated with a RPA software called UiPath [4], which is also explained in this section.
- *Design.* Here it is explained the processes that could be automatized, and the design of those automations.

- *Development*. Starting from the designs from the previous section, here it is explained how those automations were developed.
- *Test and results.* After having the automated processes, here we will test them and check that the performance of RPA is much higher than humans, as considered in the objectives.
- *Conclusions and future work.* This section was used to check if the objectives were reached, and the conclusions from all this work. Some extensions which could be added in the future are also explained.

2 State of art and technology

In this section, we will focus on the study of the different technologies used in this work. First, we will comment the Database Management System (DBMS) of Oracle Databases. It is important to understand its architecture, because we will automatize some DBMS processes afterwards. The automation technology used is the RPA software called UiPath, so we also need to understand its characteristics, potential, and weaknesses. This is also important, so we know all the options that this tool gives us.

2.1 Oracle Database

Regarding the Database Administration, there are several aspects worth knowing before getting into the automation of some tasks inside the Database. First of all, we studied the architecture of the DBMS, and how an Oracle Database works. After that we will see other aspects, such as the user administration and the concurrency and consistency of data. And, finally, some Database Administrator (DBA) common tasks, which will be automatized with the RPA [5].

Regarding its architecture, we will focus on three main areas of every database. They are: memory architecture, process architecture, and disk architecture, with its physical and logical structures [5].

2.1.1 Memory architecture

As explained on the Oracle Databases Documentation [5], there are three memory structures, and they are provided for different purposes:

- **System Global Area (SGA).** It is a part of memory which stores the Oracle Database instance. Therefore, it contains all data and control information of the instance. It is a shared memory, so its information can be accessed by different users. The SGA is also divided into different components. The most important ones are:
    - *Database Buffer Cache*. It is a buffer which contains copies of data blocks from data files previously read.
    - *Redo Log Buffer*. It is a buffer with enough information for redo changes. It is provided because it could be necessary to undo changes in the database.
    - *Shared Pool.* It is a shared memory for storing the dictionary cache, the library cache, the result cache, control structures and buffers for parallel execution messages.
    - *Large Pool*. It is a memory area prepared by a DBA to store big memory allocations such as back-ups.

- *Java Pool*. It stores all Java code and information from all users inside a JVM.
- *Streams Pool*. It is used only for Oracle Streams. Its size increases dynamically while the stream is used.

- **Program Global Area (PGA).** This part stores the server processes. There will be one PGA for each created process, and they are not shared. This means that only the server process itself will have access to it. Inside the PGA, there is two different areas:
  - *Session Memory*. It keeps the information about a session.
  - *Private SQL Area*. It contains information about the executed queries in such process. For example, the value of variables and the state information of queries.

- **Software code areas**: This is the place where all the code is stored. The one which is running, and the one which can be run. It is located in a more protected area than the user programs.

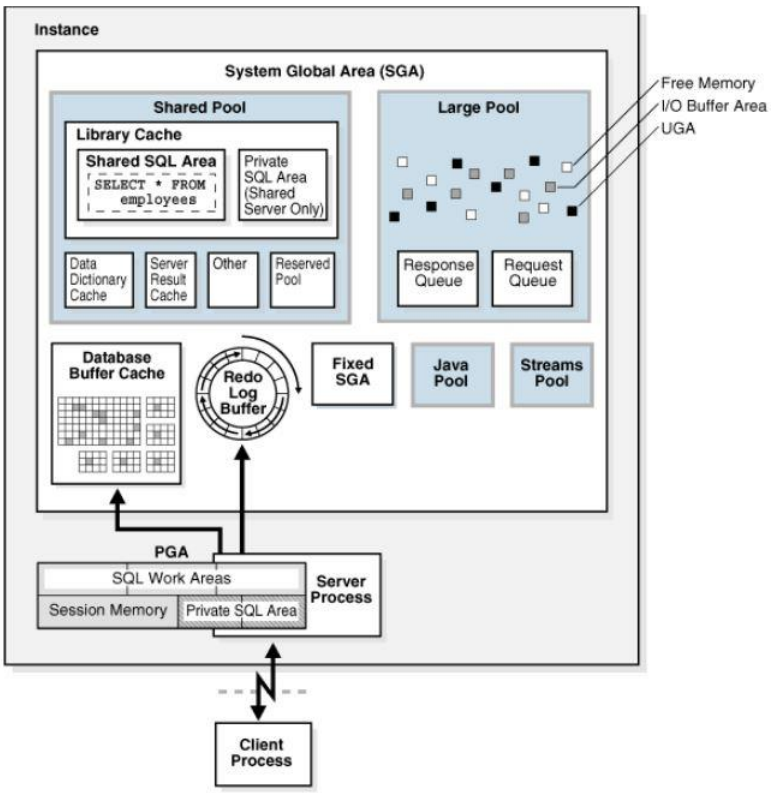


Figure 2-1: Memory structure of an Oracle Database

2.1.2 Process architecture

First of all, let´s see what we understand by process. A process is a sequence of instructions run in an Operating System. As explained on [6], in order for a user to access the instance of an Oracle database, two modules of code should be run:

- *Application or Oracle Tool*. Which works as an intermediary for the user to send the SQL commands to the database.
- *Oracle database server code*. They are the processes which receive and execute the SQL queries sent by the user.

There also exists Multiple-Process Oracle Systems. They are also known as multiuser Oracle, and they allow multiple users to work at the same time on a single database instance. [7]

There are two types of processes inside an Oracle database [7]:

- *User processes*. They are created by the database for the user to work on the application or Oracle tool. This type of process allows a user to connect to the Database. Once the connection is correctly established, a session is opened for him or her to interact with the database [7].
- *Oracle Database processes*. There are three different types of Oracle Database processes [7]:
    - *Server processes*. They are on the other side of the connections previously mentioned. The Oracle Database creates these type of processes to get and perform the desired requests from users. These types of processes usually consist of: receive the queries, execute the desired action on the database, and return the results from those actions.
    - *Background processes*. They are automatic processes in charge of keeping the good efficiency of the database. Some of the most important background processes are:
        - *Archiver Processes (ARCc).*
        - *Checkpoint Process (CKPT).*
        - *Database Writer Process (DBWn).*
        - *Job Queue Processes.*
        - *Log Writer Process (LGWR).*
        - *Process Monitor Process (PMON).*
        - *Queue Monitor Processes (QMNn).*
        - *Recoverer Process (RECO).*
        - *System Monitor Process (SMON).*

2.1.3 Physical database structures

As explained on [7], there are several structures:

- *Data files*. They are the files where all data are stored physically. Therefore, even data from logical structures are saved into data files. A collection of data files forms logical structures, which are called tablespaces.
- *Control files*. There is at least one for each Oracle Database. It has important information about the physical structures, such as the locations of the data files and Redo Log Files.
- *Online Redo Log Files.* They contain the changes performed to data, so undo operations can be done.
- *Archived Redo Log Files*. They are offline copies of the previous one.
- *Parameter Files*. They save the initial parameters chosen for the database and the instance.
- *Alert and Trace Log Files.*
- *Back-up File.*

2.1.4 Logical database structures

As explained on [8], logical database structures help the Oracle Database to control the use of disk space, and to isolate the physical from the logical storage. From smaller to larger, they are:

- *Oracle Database Data Blocks*. They contain bytes of data from data files. It is the smallest size of logical structures. It is the structure in which the I/O operations of the Database are transferred.
- *Extents*. Collection of adjacent Data Blocks.
- *Segments*. It is a group of Extents.
- *Tablespaces*. They contain all the previous logical structures, and they are used to simplify DBA tasks. All Oracle Databases have two default tablespaces which cannot be removed, SYSTEM and SYSAUX. The SYSTEM tablespace is the main tablespace from every Database, and stored important information about the database server. The SYSAUX tablespace is an auxiliary tablespace. “Each tablespace in an Oracle database consists of one or more files called data files” [8].

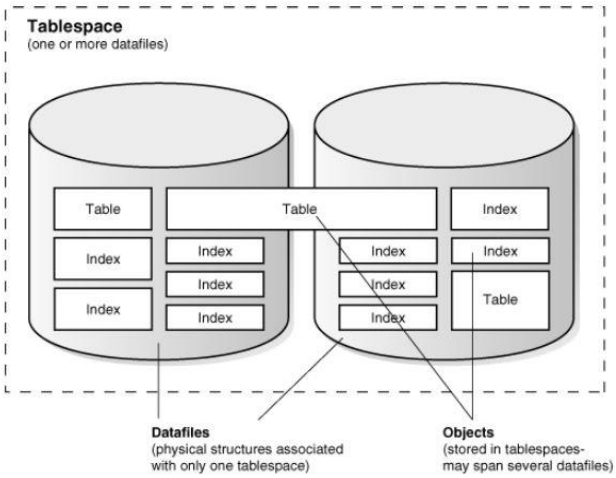


Figure 2-2: "Datafiles and Tablespaces" [9]

2.1.5 User management

As explained in [10], Oracle Database allows the creation of many users. Each of them will have a username and password, which are necessary to establish a connection to the database. Every user will have specific permissions for:

- Tasks that they can perform in order to alter the state of the database, like the content of tables.
- Accessing different areas (tablespaces).
- Using a specific amount of the resources of the database, like the CPU processing time.

All those permissions to users can be organized with roles. Roles can be created and assigned to users, so user permissions are easily managed by giving them to specific roles, instead of granting them to all of them one by one.

Regarding the disk areas they have access to a default tablespace that is assigned to every user when it is created. If they are granted with the right permissions, users can also access tables from other tablespaces. These tablespaces, together with the tablespace quotas, limits the disk space that the user will have access to. Apart from that, when a user is created, he will also have a temporary tablespace, which will be used for certain SQL operations that need a temporary space such as sorting.

2.1.6 Concurrency, consistency and locks

One of the most important characteristics of an Oracle Database is to allow concurrency among users. And, as explained in [11], that allows multiple users to access the same data at the same time. Consistency is also a high priority for an Oracle Database, so there is no

difference between data read by several users at the same time. It is also important that there are not several writing operations over the same data at the same time.

Read consistency should guarantee:

- A reader does not wait for other readers or writers to finish, and the read data is not changed during the reading.
- A writer needs to wait for other writers to finish and commit their work. This way, any update is always done based on updated data.

An Oracle Database achieves that due to transactions, providing lock operations over data and a multi-version consistency model. This model works as if every user had a copy of the database, and that copy is updated whenever a user finishes and commits a transaction which updates de data from the database.

Locks are a very important part of an Oracle Database, and it is the developers' duty to use them correctly. Through this tool, resources are locked while they are being updated by a user transaction. This avoids the case of two different users updating the same data simultaneously. When a second user tries to update a data which is already in a transaction, the process is locked until the first user has committed his work.

2.1.7 DBA common tasks

In [12] and [13] there is some of the common tasks that a DBA is responsible of are:

- Manage storage, both physical and logical structures of the database, like data files, control files, tablespaces... Since the creation and until the deletion of the instance of the database.
- Manage users and roles. Create them, modify their quotas, administrate the permissions, and delete them.
- Look for possible conflicts and solve them. For example, look for permanent locks in resources of the database and free them if they are not being used.
- Configure back-ups and have a strategy to solve issues which may happen to the database.
- Stay up to date with the latest updates and versions of software. Also educating other DBAs and developers.

2.1.8 Robotic Process Automation

Robotic Process Automation (RPA) is a technology which combines Artificial Intelligence (AI) and the automation of processes. They can work like a human does, by moving the mouse, typing and using shortcuts, looking and waiting for User Interface (UI) elements to appear on the screen… [14] Before this technology, automation was done through Application Programming Interfaces (APIs), and the real innovation of RPA is that it allows automation based on the UI.

The goal of RPA is to substitute humans on large and repetitive tasks. This way, we can have these tedious processes done by computers. RPA is faster, cheaper, and more efficient and they allow humans to focus on more complex tasks [15]. To summarize, RPA is better than people with structured processes because they usually do not fail as many times as humans fail. Of course, this happens when they are well programmed.

A RPA does not learn by itself. It needs to be trained and taught by rules about how to act in every possible scenario it can face. And it is the developer´s responsibility to program the RPA for that.

Currently, RPA are specially used for tasks like the following ones [16]:

- Activating buttons.
- Filling forms with data.
- Extracting and organizing data from forms.
- Gathering data. For example, storing data received from many emails.

Thanks to all those features, RPA is getting more and more popular among companies inside the domains: finances, database administration, business management applications like SAP, customer service… [16]

Some of the most popular RPA vendors are [17]:

- **UiPath:** it is simply and great for RPA beginners. It provides a lot of education, from YouTube videos to the UiPath Academy, which is also for free. And these are the reasons why it was the selected tool for this work.
- **Blue Prism:** it is prepared for big projects and companies. It has a wider range of resources and tools, but their licenses and academy are very costly.
- **Automation Anywhere:** it is made for experienced RPA developers, and who want to have more control over the automation options.

2.1.9 UiPath

Because of the reasons previously explained, this was the chosen software for the automation. This vendor provides different products [18]:

- *UiPath Studio*: it is used to design the automations without coding. This is the only development tool used for this work.
- *UiPath Orchestrator:* it is a browser-based server application used to manage and monitor big projects with multiple automations, focused on big companies
- *UiPath Robot*: it executes the processes UiPath Orchestrator.

Development using UiPath Studio is very visual. It is done by using activities to perform each action, instead of coding. Some examples of these activities are [19]:

- *Open Application.*
- *Assign. To store a value in a variable*.
- *Click and Double Click.*
- *Type Into.*
- *Send Hotkey.*
- *On Element Appear/Vanish.*
- *Get Full Text.*
- *Get Optical Character Recognition (OCR) Text.*
- *Build Data Table.*
- *Read Range (from an Excel file or workbook).*

UiPath Studio allows users to create projects for each automation of rule-based processes. And there are three types of projects [20]:

- *Sequences:* for linear processes.
- *Flowcharts:* for more complex processes.
- *State Machines:* for bigger projects.

UiPath Studio has many common features with any programming language. Some of them are: [20]

- Use of variables. They are used to store data and they allow all of the usual types (generic value, integer, double, char, string, data table, time and date, different types of collections…).
- It allows to extract project as workflows and use them as methods, with input/output arguments.
- For the control flow, we have activities for different conditionals and loops.

As an RPA, it has some other special tools, and they are explained in the official documentation of UiPath [20]:

- *Selectors*. "These store the attributes of a graphical user interface element and its parents, in the shape of an XML fragment. Most of the times, selectors are automatically generated by UiPath Studio and do not require further input from you, especially if the apps you are trying to automate have a static user interface." [21] But it is also possible to edit them manually.

- *Data Scraping.* “Data scraping enables you to extract structured data from your browser, application or document to a database, .csv file or even Excel spreadsheet” [22].
- *Image and text recognition*. This is done by using the technology called Optical Character Recognition (OCR). The screen is recorded and the technology scraps the images and text. It is used when selectors do not work properly in virtual machine environments [23].

3 Design

Now that we have understood the architecture of Oracle Databases, and the possibilities that RPA gives us for automating processes, it is time to study how they can be combined. This work is focused on three repetitive tasks, which are frequently done by DBAs, and they have been automated with UiPath. The development environment SQL Developer was used to work with the Oracle Database. The three processes this work automates are:

- **Creation and drop of users**. This process is the first one I automated because it was the simplest. It may not seem very useful to automate this, but it can save a lot of time if the DBA needs to create or delete many users.
- **Creation and drop of tablespaces**. This is done by DBAs to manage the logical storage of the database. This case, just like the previous one, can save a lot of time if many tablespaces need to be created or deleted. In operative environment sometimes appear critical situations where a new tablespace needs to be created.
- **Free locked resources.** We have seen before how users can lock Oracle Database resources. The problem is that they are sometimes kept locked unintentionally for a long time, and this stops other users from using those resources. With this automation, those locked resources can be freed automatically by the RPA.

In the following sections we will see in detail how these tasks are currently done visually (with UI elements on SQL Developer), by DBAs typing the necessary queries to perform such tasks, and a proposal design for how those tasks could be performed by RPA instead of administrators. This can save a lot of time to technicians.

3.1 Use Case 1: Creation and drop of users

3.1.1 Creation of users

This use case automates the process of creating users. To create a user, only the username and password are needed, but we will also add the option to choose its default tablespace and temporary tablespace. Oracle SQL Developer allows us to create a user in two different ways:

- Visually. This is done through a window which appear after a right click on the list of users, and selecting the option “Create User”. Next image shows how the window looks after typing the username, password, and default and temporary tablespaces. The user will be created after clicking on “Apply”.

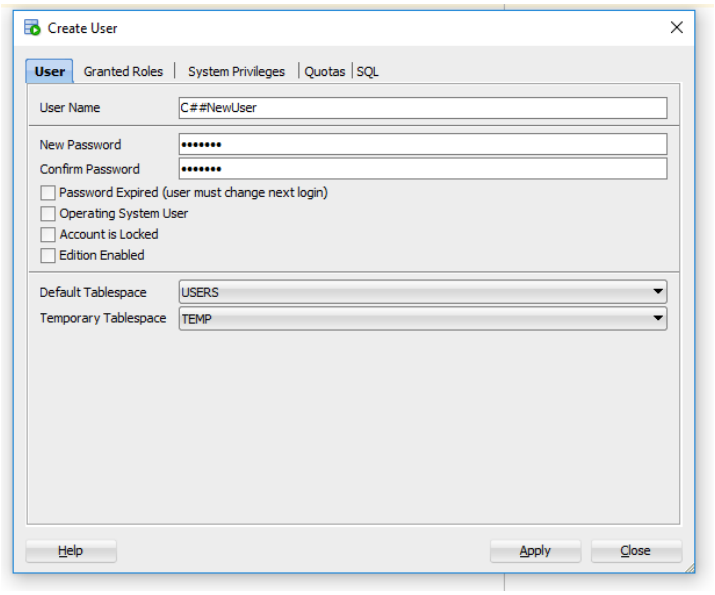


Figure 3-1: SQL Developer window for creating a user

- By typing the query in the SQL Worksheet. For creating a user like the one above, we would need to type the following query and execute it:

  CREATE USER C##NewUser IDENTIFY BY NewUser
  DEFAULT TABLESPACE "USERS"
  TEMPORARY TABLESPACE "TEMP";

Both ways allow us to create a new user for the database. But if we want to create multiple user, this process would take a long time. We can make it faster and simpler by letting the DBA a table with the users to be created, with their data. The table would look like the following one, and we can let the RPA do the rest.

Username	Password	Default tablespace	Temporary tablespace
NewUser	Newuser	USERS	TEMP
NewUser2	Newuser2	USERS	TEMP
…	…	…	…

Table 3-1: Scheme of the Excel file to create users

The automation process would read the users´ data from an Excel file with that structure, build the queries and execute them one by one automatically. This way, we can have an automated process doing the work as a human would do. This diagram represents the process:

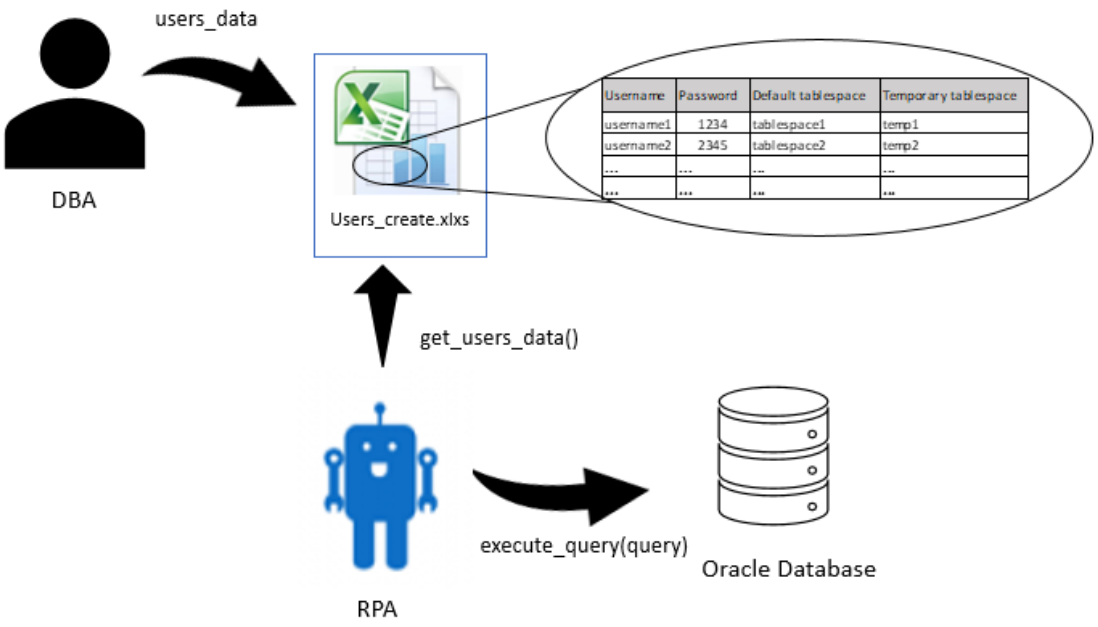


Figure 3-2: Diagram of the creation of users

3.1.2 Drop of users

This use case complements the previous one, but it is simpler. To delete a user from an Oracle Database, the DBA just needs the username and decide about the "cascade" options, which would also delete all the user´s objects.

Oracle SQL Developer allows you to delete users in three different ways:

- By opening the list of users, and doing a right click on the target user, and selecting the "Drop User". This opens a window like this one, in which we can select the "Cascade" option:

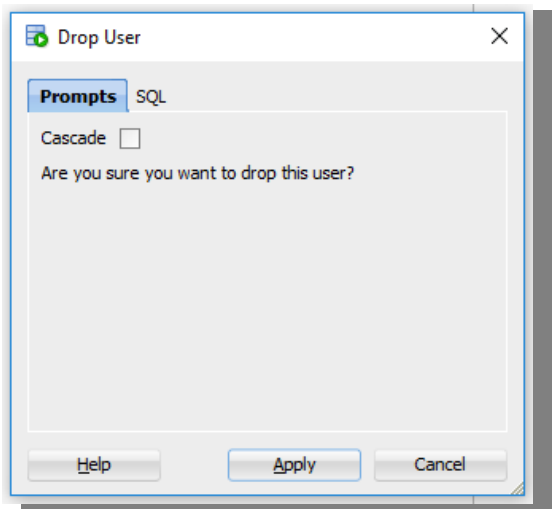


Figure 3-3: SQL Developer window for dropping a user

- By opening the list of users, and doing a right click on the list of users, and selecting the option “Drop user”. This will open a window in which we will have to select the username that we want to drop, and select the “Cascade” option if we want to. It would look like this if we want to delete “OldUser” with the Cascade option.

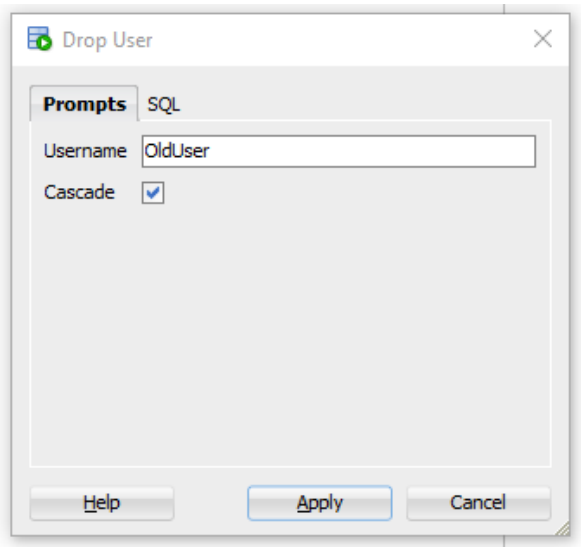


Figure 3-4: SQL Developer window for dropping any user

- And the third one, is by writing and executing the query to drop the user on the SQL Worksheet. For deleting a user like above, we would have to type and execute the following query:

  DROP USER “C##OldUser” CASCADE;

These three ways allow us to delete a user from the database. But, just like in the previous use case, it would take us a long time to delete multiple users. So, we can make it faster and simpler by providing the DBA with and Excel file with the following structure, so he is able to delete many users at one time with UiPath:

Username	Cascade Option
OldUser	Yes
OldUser2	No
…	…

Table 3-2: Scheme of the Excel file to drop users

The automation process would read the usernames from the file, build the queries and execute them one by one automatically. This way, we can have an automated process doing the work as a human would do. This diagram represents the process:

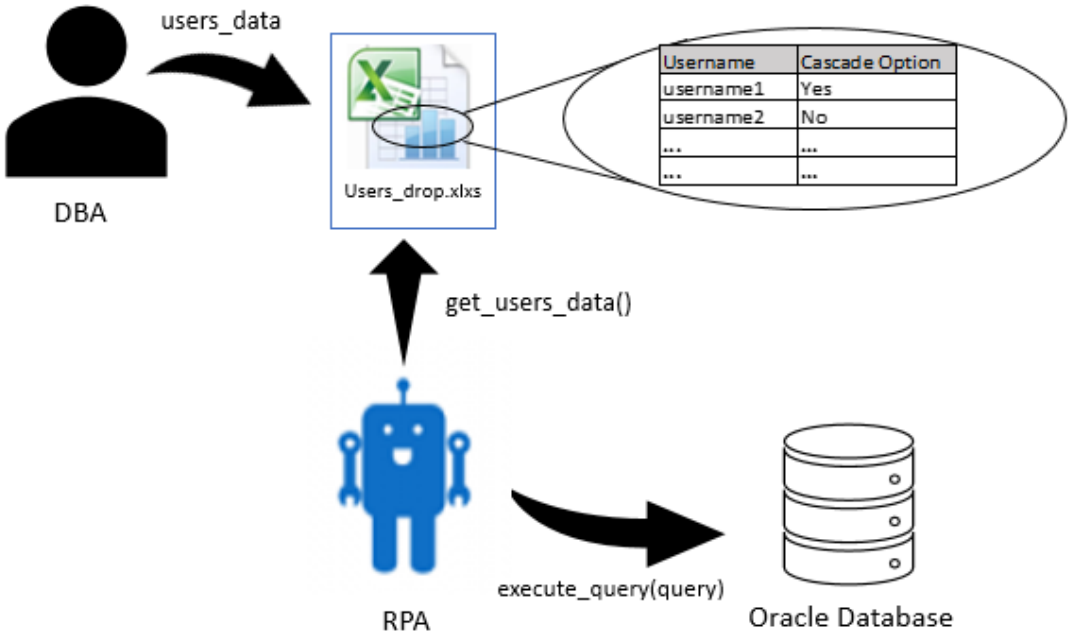


Figure 3-5: Diagram of the drop of users

3.2 Use Case 2: Creation and drop of tablespaces

3.2.1 Creation of tablespaces

In this section, we study the automation of the process of creating tablespaces. This task focuses on the process of managing logical storage of any Oracle Database. But before automating it, we need to understand how the process is manually done by a DBA. The creation of tablespaces is a more complex process than the creation of a user, because we need to specify several features:

- Tablespace name. It is unique.
- Tablespace type. It can be permanent, temporary or undo.
- Data file name. This is where data will be physically stored.
- Data file size and unit of measure. The size will be a number, which can be in KB, MB, GB or TB.
- Auto extend on. Option which allows automatic extensions of the tablespace data file.
- Next size and unit of measure. The size of each extension with its unit, which can be KB, MB, GB or TB.
- Max size and unit of measure. This will be the maximum size of the data file.

- Other properties, which will be set by default by the RPA for permanent and undo tablespaces, such as:
    - Directory: this is where the data files will be stored.
    - File type: Small file.
    - Online Status: Online.
    - Block size: auto.
    - Logging: No logging.
    - Segment management: Auto.
    - Auto allocate: True.

Just like with the creation of users, we can create tablespaces in several ways. The main ones are:

- Visually: Through a window which opens after a right click in the tablespaces list, and clicking on “Create tablespace”. A window opens, where we have to decide about the new tablespace features. It has several tabs, but we will leave the default options, except on the first one. It will look like this:

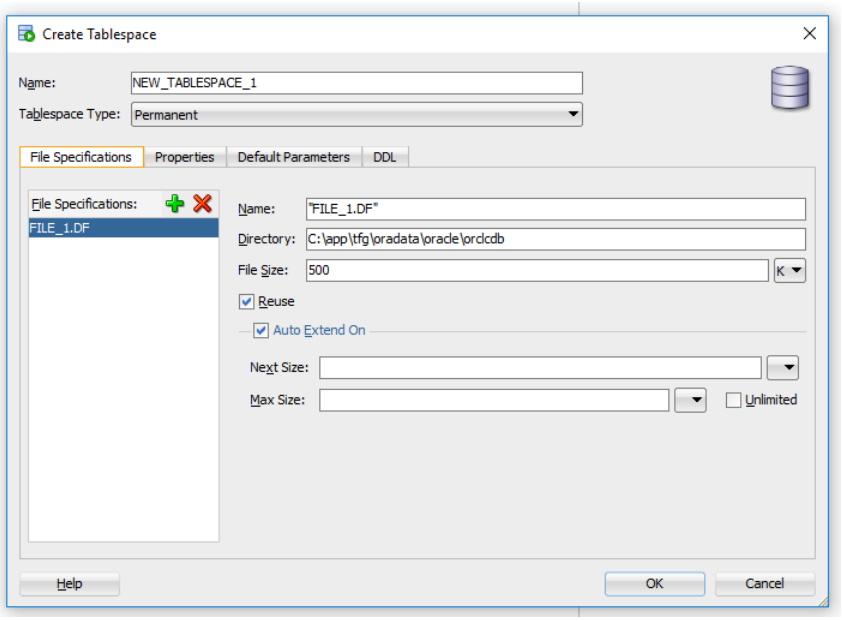


Figure 3-6: SQL Developer window for creating a tablespace (Example 1)

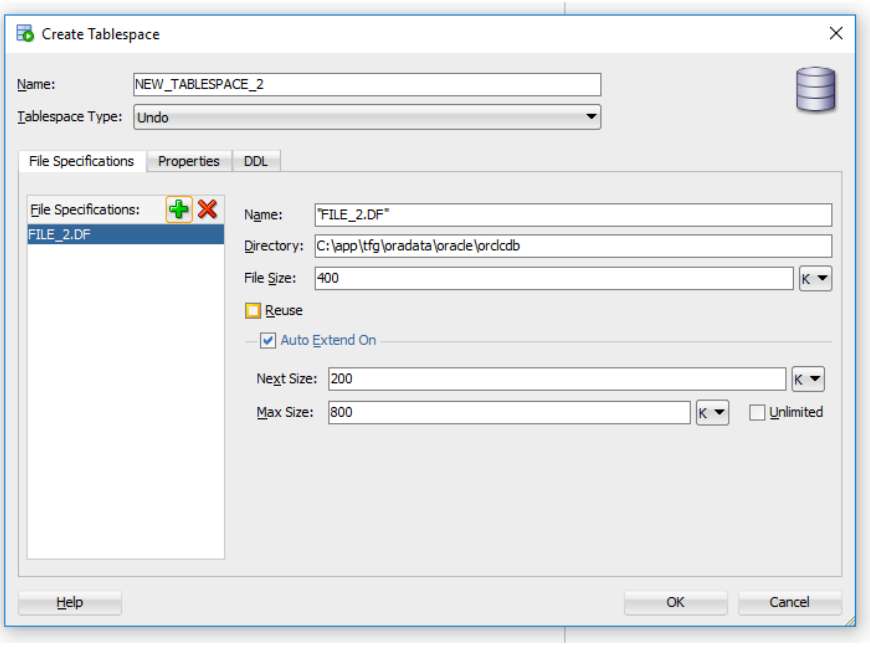


Figure 3-7: SQL Developer window for creating a tablespace (Example 2)

- By typing and executing the query manually. This way it is a little more complex the creation of a user, because there are more features to be chosen. The queries for the previous tablespaces would be:

```
CREATE SMALLFILE TABLESPACE NEW_TABLESPACE_1
   DATA FILE
      'C:\app\tfg\oradata\oracle\orclcdb\FILE_1.DF' SIZE 512000 REUSE
AUTOEXTEND ON
   NOLOGGING
   ONLINE
   SEGMENT SPACE MANAGEMENT AUTO
   EXTENT MANAGEMENT LOCAL AUTOALLOCATE;

CREATE SMALLFILE UNDO TABLESPACE NEW_TABLESPACE_2
   DATA FILE
      'C:\app\tfg\oradata\oracle\orclcdb\FILE_2.DF' SIZE 409600 AUTOEXTEND
ON NEXT 204800 MAXSIZE 819200
   EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

We can create tablespaces with any of those options, but it is a repetitive and tedious task if we want to create many tablespaces. Our goal, is to automatize it with RPA, so we just have to complete the desired features in a table, located in an Excel file. It would look like this:

Tablespace info		Data file							
		Basic info			Auto extend				
Name	Type	Data file name	Size	Unit	Activate autoextend	Next size	Unit	Max. size	Unit
NEW_ TABLESPACE_1	Permanent	FILE1.DF	500	K	Auto				
NEW _TABLESPACE_2	Undo	FILE2.DF	400	K	Yes	200	K	800	K

Table 3-3: Scheme of the Excel file to create tablespaces

To automate this process the following the steps were addressed:

1. DBA writes the tablespaces data that he wants to create in the Excel file called "tablespaces_create.xlxs".
2. The RPA reads data from Excel file "tablespaces_create.xlxs".
3. The RPA builds the queries and execute them on SQL Developer.
4. The RPA writes a copy of those created tablespaces in the file "tablespaces_existing.xlxs". This step will be helpful for the following use case, which is about deleting tablespaces, so the users don't have the option to delete non-existing tablespaces.

These steps are represented in the following diagram:

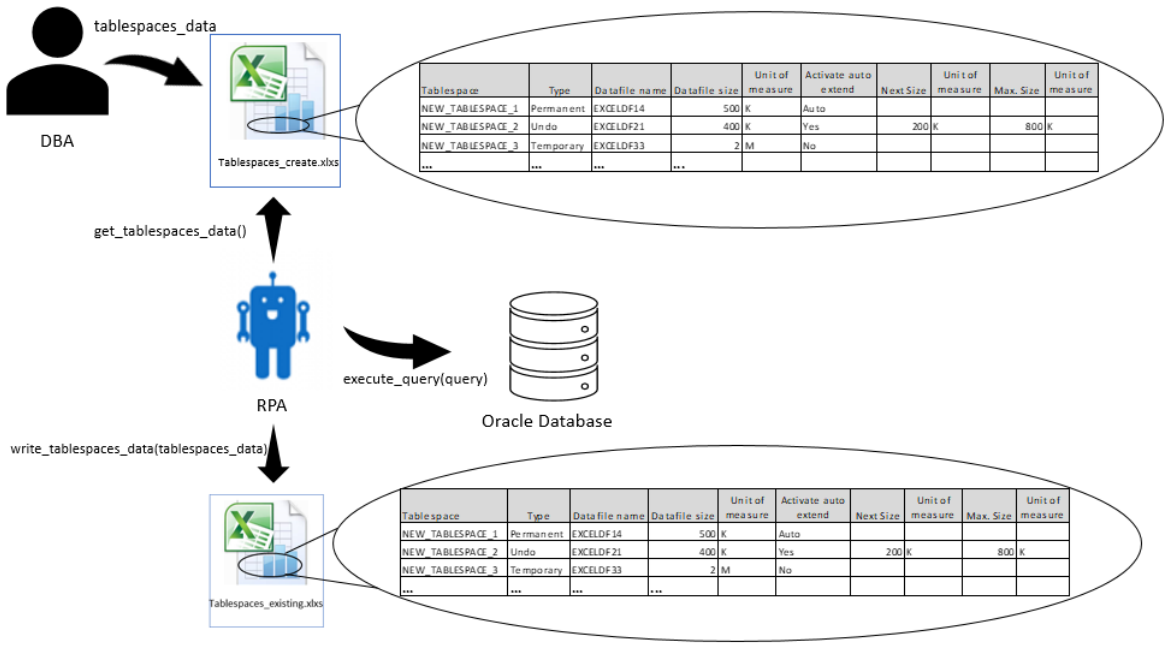


Figure 3-8: Diagram of the creation of tablespaces

3.2.2 Drop of tablespaces

In this use case we study how DBAs drop tablespaces. It complements the previous use case, which creates them. To delete a tablespace, it is necessary to select its name, and decide if to activate the following options:

- Include contents.
- Include data files.
- Cascade constraints.

As usual in SQL Developer, this can be done in two main ways:

- Visually, by selecting the tablespace, and then the option “Drop tablespace”, and then select the desired options in the window which appears. This image shows that window:

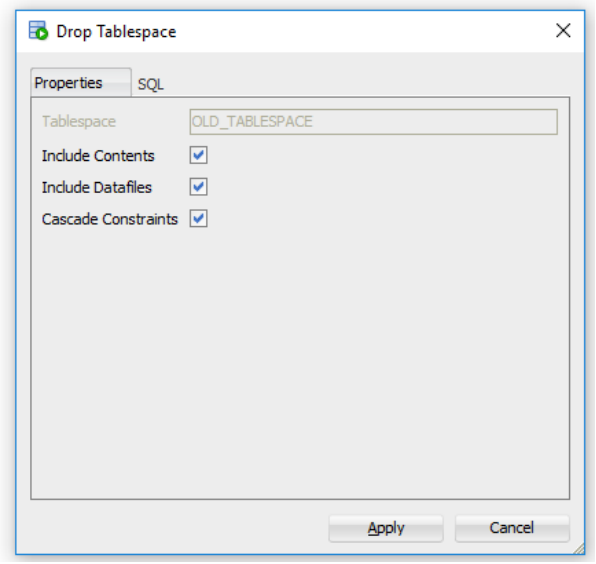


Figure 3-9: SQL Developer window for dropping a tablespace

- Manually, by typing and executing the query on the SQL Worksheet. For the tablespace from before, we would have to type the following command:

  DROP TABLESPACE "OLD_TABLESPACE"
  INCLUDING CONTENTS AND DATA FILES
  CASCADE CONSTRAINTS

These two ways allow us to delete a tablespace from the database. But we can make it faster and simpler by providing the DBA with and Excel file with the following structure, so he is able to drop many tablespaces at one time with UiPath:

Tablespace name
OLD_TABLESPACE
…

Table 3-4: Scheme of the Excel file to drop tablespaces

Once again, the more tablespaces we want to drop, the more time we can save by using RPA. We can also add a control to check that the tablespace we want to drop already exists, and this can be done by keeping a copy of the existing tablespaces in an Excel file. The steps followed to automatize this process would be:

1. The DBA writes the names of the tablespaces he wants to create in an Excel file called "tablespaces_drop.xlxs". In this file, we control that only existing tablespaces can be dropped. We do this by comparing the typed tablespaces with the ones in the file "tablespaces_existing.xlxs".
2. The RPA reads the data from that file.
3. The RPA builds the SQL commands and executes them one by one in SQL Developer.
4. In order to keep track of the existing tablespaces, the dropped tablespaces are deleted from the file "tablespaces_existing.xlxs".

This process is represented in the following diagram:

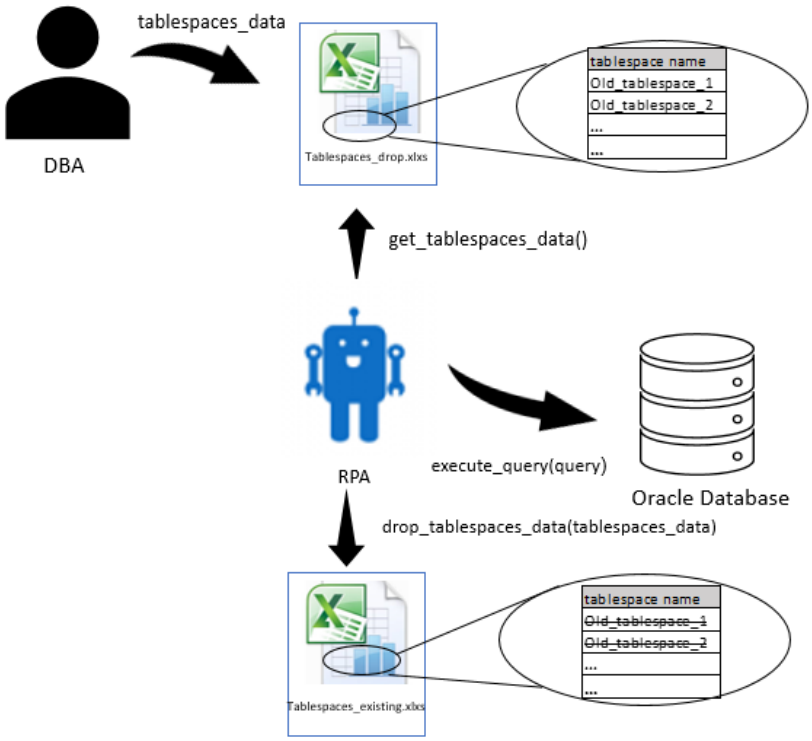


Figure 3-10: Diagram of the drop of tablespaces

3.3 Use Case 3: Free of locked resources

This use case is the most complicated, but probably the most useful for DBAs. It is very common in Oracle Databases to find locked resources. When this happens, the DBA needs to check the locked resources and the sessions locking them, and kill the inactive sessions locking resources [25].

The only possible way to do this is by executing the appropriate queries for getting the information about the sessions locking resources, and kill the inactive ones afterwards one by one. As you can imagine, this a tedious process that can be automatized with RPA. [25]

Any kind of resources can be freed this way, but we will focus on the example of tables. They are locked during the process of a user changing its data. It can be done through transactions, which allow users to alter data from tables. When a user starts a transaction and modifies the content of a table, that content of the table keeps locked until the work is committed. While having locked content in a table, any user trying to lock that same content will have their session blocked, until the first user commits his work. This can lead to a situation in which there are several sessions waiting for that content of the table to be unlocked. After becoming unlocked, it becomes available and it will be locked again automatically by the first session from the waiting list [11].

It is very possible that the users who locked the resources are not active anymore, and this does not allow other users to have access to those resources, and blocks their sessions. This is the reason why we want to kill the inactive sessions which are locking resources.

This problematic situation is represented in the following diagram:

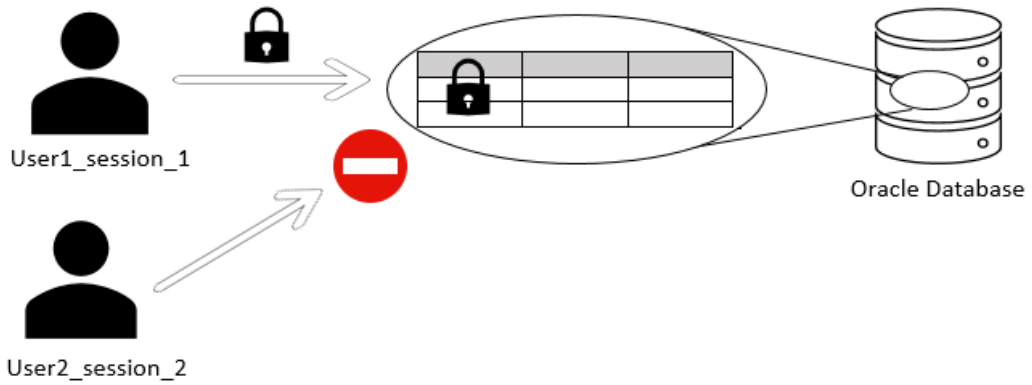


Figure 3-11: Diagram of a locked table

The required steps to solve an example of this problem, are:

1. SQL command to get the information about the sessions locking resources. This query can be modified, depending on what you want to get, but the following one gives us more than enough information.

SELECT c.OWNER, c.OBJECT_NAME, c.OBJECT_TYPE, b.USERNAME, b.SID, b.SERIAL#, b.STATUS FROM v$locked_object a, v$session b, dba_objects c WHERE b.SID = a.SESSION_ID AND a.OBJECT_ID = c.OBJECT_ID AND b.STATUS = 'INACTIVE'; [24]

And the result we get when one user (USER2) is locking a table (AAATABLE) is the following table:

OWNER	OBJECT_NAME	OBJECT_TYPE	USERNAME	SID	SERIAL#	STATUS
USER1	AAATABLE	TABLE	USER2	378	22902	INACTIVE

Table 3-5: Scheme of the Excel file with information about the locked resources

2. Execute the SQL commands to kill all the inactive sessions locking those resources. For that, we need the "SID" and "SERIAL#" of the target sessions. In the previous example, the query would be:

   ALTER SYSTEM KILL SESSION '378,22902';

After this, the session locking the table is killed, and the content of the table becomes available for the first user who requested access to it.

But, if there are many locked resources, this would be a much longer task. That is why automating this process is very valuable. The steps for this to be done by an RPA are very similar to how it is currently done by DBAs, with the only difference of using an Excel file to store the information about locked resources and sessions. Using RPA, a DBA will not even have to be present to solve this, the RPA could solve it by itself. And this is another reason why RPA is very useful.

This process is represented in the following diagram:

1. The RPA executes a query to get the information about resources and sessions locking them. The result of the query is stored in the Excel file "Locks.xlxs".
2. The RPA get the data of the locked resources and sessions from the Excel file.
3. The RPA builds and executes the queries to kill those inactive sessions locking resources to make the resource available again. This image shows that process:

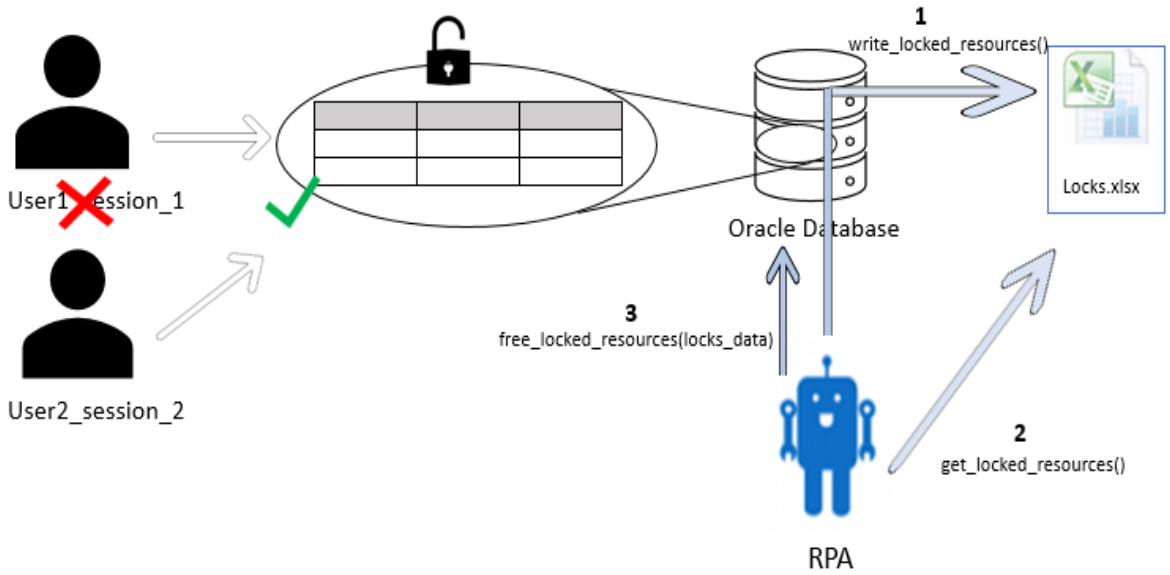


Figure 3-12: Diagram of the freeing a locked table

Once again, having a process automated can save us time, and with this one we do not need a DBA executing queries. Anyone could just run the UiPath project, and the RPA does the rest.

4 Development

Most of the development was done in the RPA platform called UiPath. This RPA was reading and writing data from Excel files and executing queries in Oracle SQL Developer, which is user for managing Oracle databases.

In two of the use cases, there was a control of errors added in the Excel files with the functions they provide. This was done to avoid not expected data types to arrive to UiPath while reading those files and storing data into variables.

In all of the use cases, the development is explained following this structure:

- Image of the diagram with the proposed design from the previous chapter.
- Image of the main file developed in UiPath, with the pseudo code and brief explanation next to it.
- Explanation of specific workflows invoked by the main process.

Before running any of these automations, some prerequisites must be fulfilled:

- Having Oracle SQL Developer installed with a database instance created.
- Having Oracle SQL Developer opened, with an active connection, and a SQL Worksheet opened. This can be done with UiPath every time we run a program, but it is better to choose this as the starting point for the automations.

4.1 Use Case 1: Creation and drop of users

4.1.1 Creation of users

This is the diagram of the use case:

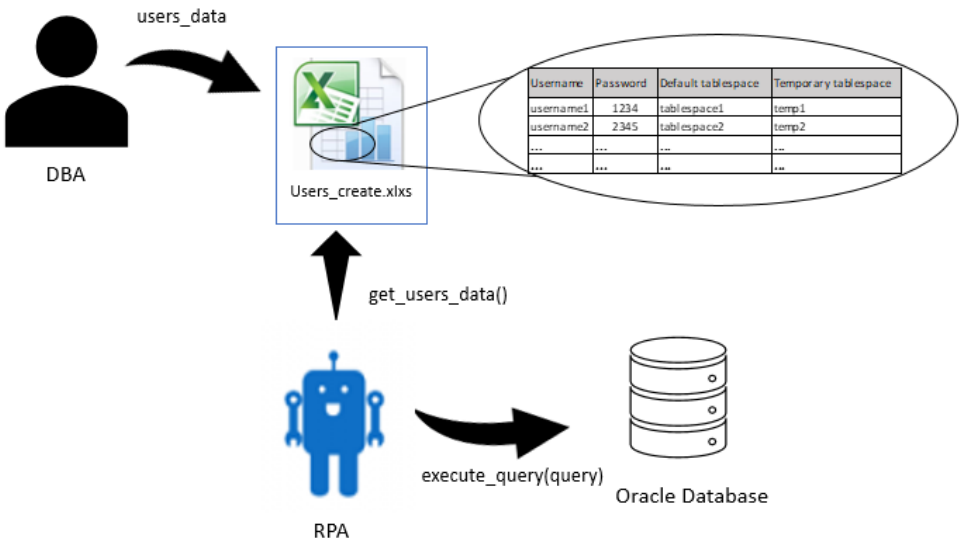
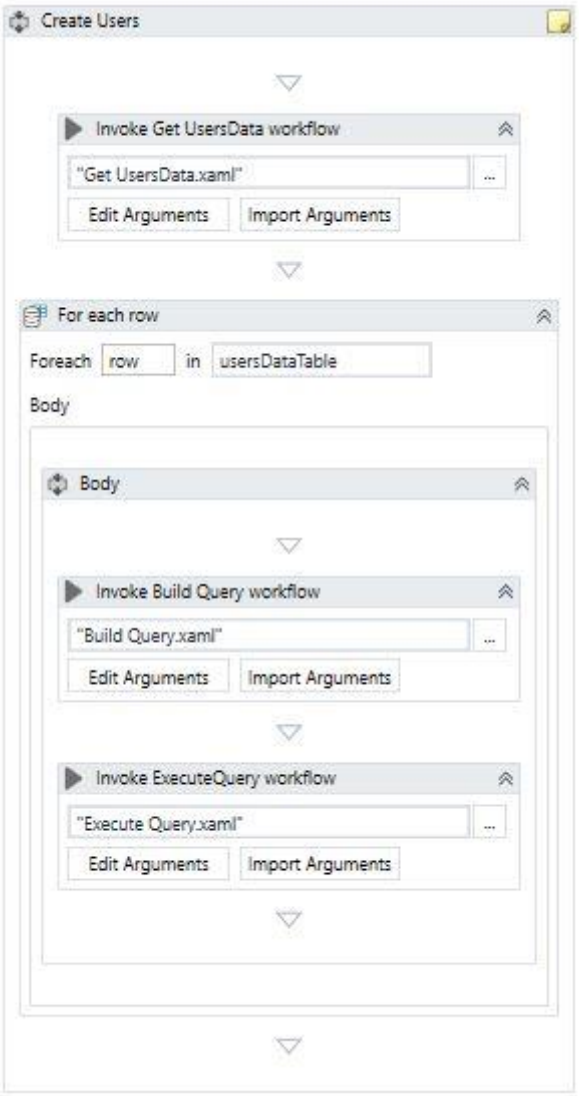


Figure 4-1: Diagram of the creation of users

This is the main file of the project, with the pseudocode and explanation of specific workflows:



Create Users:

- usersDataTable = GetUsersData().
- For each row in usersDataTable:
  - query = BuilQuery(row)
  - ExecuteQuery(query)

Figure 4-2: Automation of create users in UiPath

Get UsersData.xaml:

It is a workflow which reads the username and password of the users that are going to be created from an Excel file. It stores them in a data table, which is returned to the main workflow.

Build Query.xaml:

It is a workflow which receives an array with the specifications of a user that is going to be created. It builds the query according to the desired options, and returns it to the main workflow.

Execute Query.xaml:

It is a workflow which receives a string as an argument, and types that string in the SQL Worksheet of SQL Developer. After typing it, a shortcut for running the query is sent to the program.

4.1.2 Drop of users

This is the diagram of the use case:

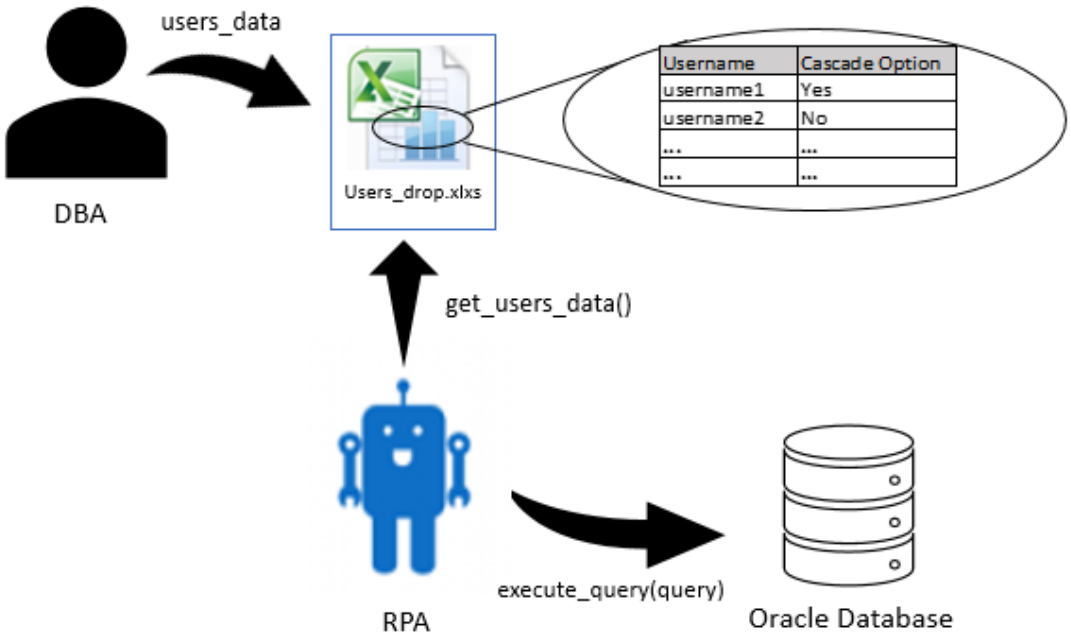
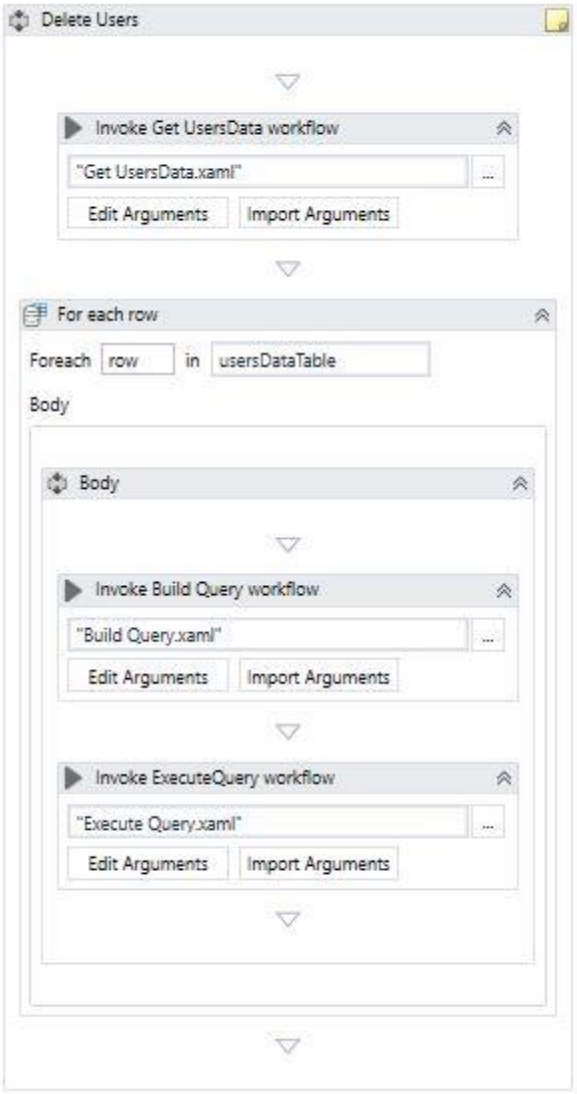


Figure 4-3: Diagram of the drop of users

This is the main file of the project, with the pseudocode and explanation of specific workflows:



Drop Users:

- usersDataTable = GetUsersData().
- For each row in usersDataTable:
  - query = BuilQuery(row)
  - ExecuteQuery(query)

Figure 4-4: Automation of drop users in UiPath

Get UsersData.xaml:

It is a workflow which reads the username and cascade option of the users that are going to be dropped from an Excel file. It stores them in a data table, which is returned to the main workflow.

Build Query.xaml:

It is a workflow which receives an array with the specifications of a user that is going to be dropped. It builds the query according to the desired options, and returns it to the main workflow.

Execute Query.xaml:

It is a workflow which receives a string as an argument, and types that string in the SQL Worksheet of SQL Developer. After typing it, a shortcut for running the query is sent to the program.

4.2 Use Case 2: Creation and drop of tablespaces

4.2.1. Creation of tablespaces

This is the diagram of the use case:

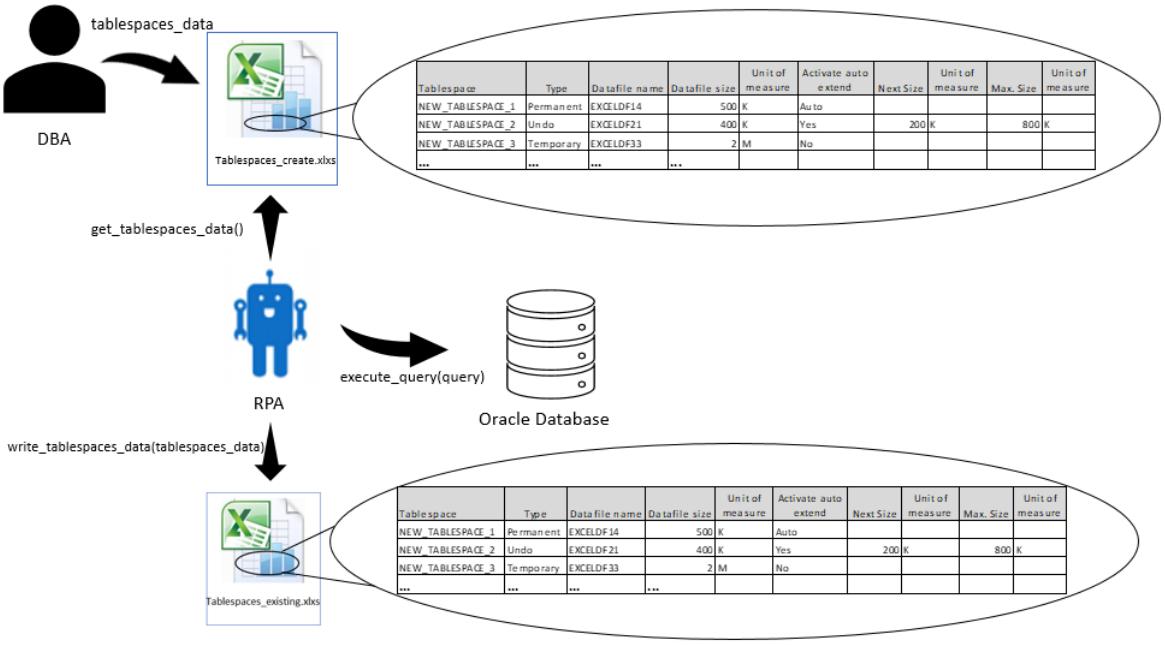
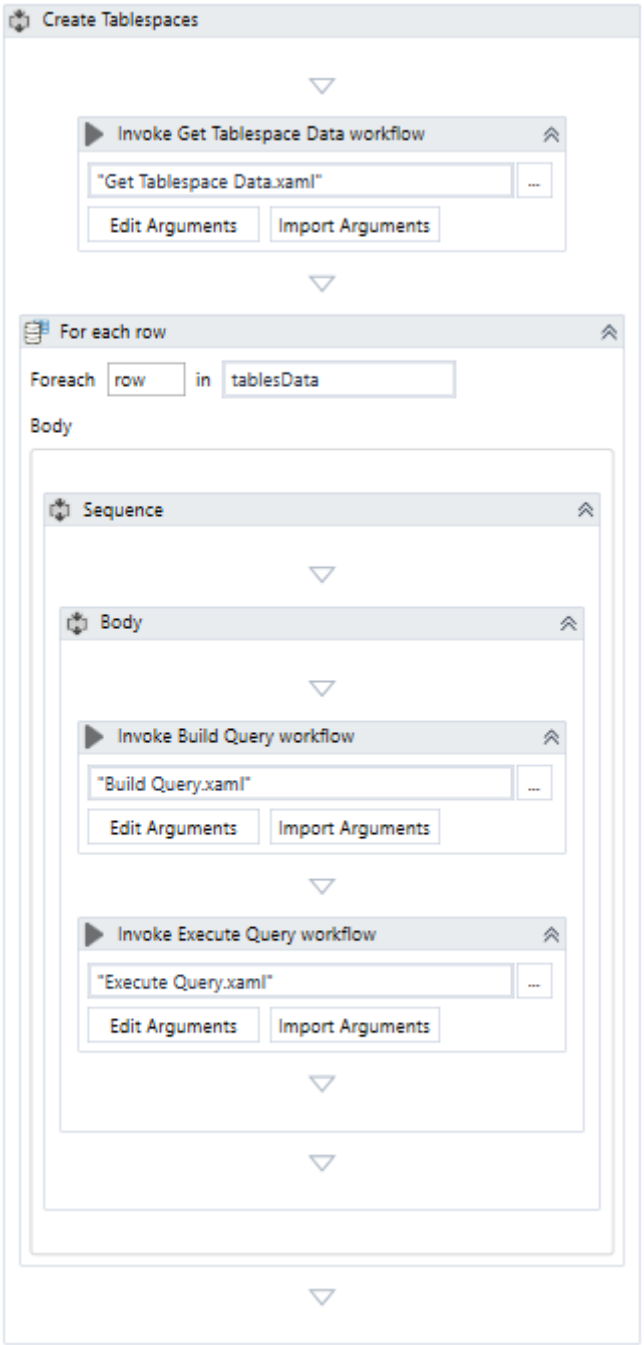


Figure 4-5: Diagram of the creation of tablespaces

This is the main file of the project, with the pseudocode and explanation of specific workflows:



Create Tablespaces:

- tablesData= Get Tablespace Data()
- For each row in tablesData:
  - query = BuildQuery(row)
  - ExecuteQuery(query)

Figure 4-6: Automation of create tablespaces in UiPath

Get Tablespaces Data.xaml:

It is a workflow which reads the specifications of the tablespaces that are going to be created from the Excel file "tablespcaes_create.xlxs". It stores them in a data table, which is returned to the main workflow. It also adds that data table to the Excel file "tablespace_existing.xlxs", which keeps track of the existing tablespaces.

Build Query.xaml:

It is a workflow which receives an array with the specifications of a tablespace that is going to be created. It builds the query according to the desired features, and returns it to the main workflow.

Execute Query.xaml:

It is a workflow which receives a string as an argument, and types that string in the SQL Worksheet of SQL Developer. After typing it, a shortcut for running the query is sent to the program.

4.2.1 Drop of tablespaces

This is the diagram of the use case:

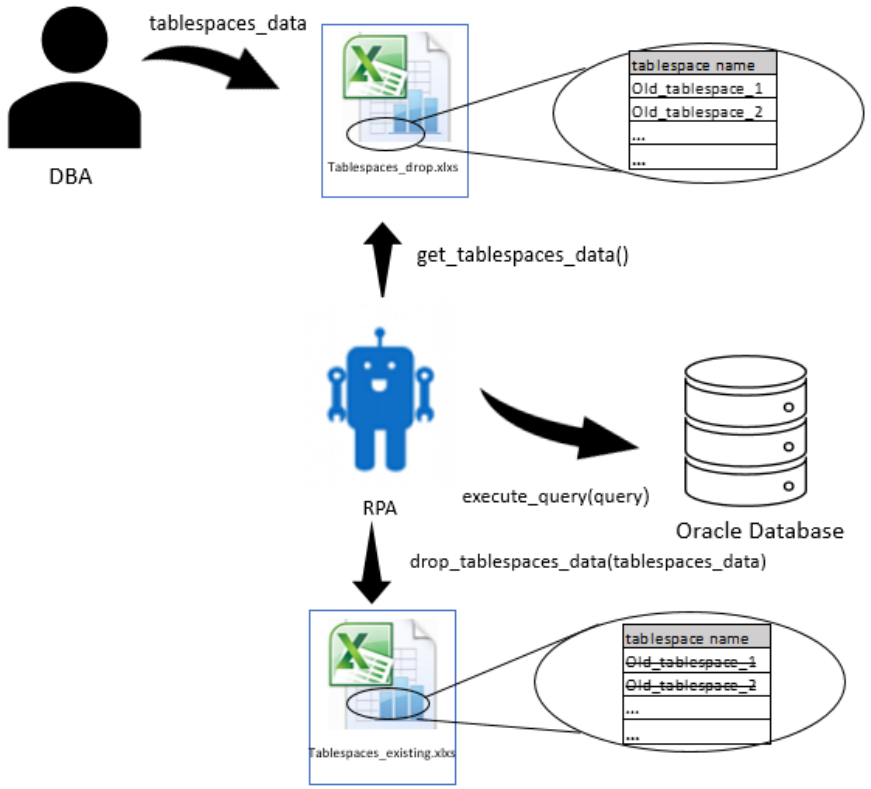
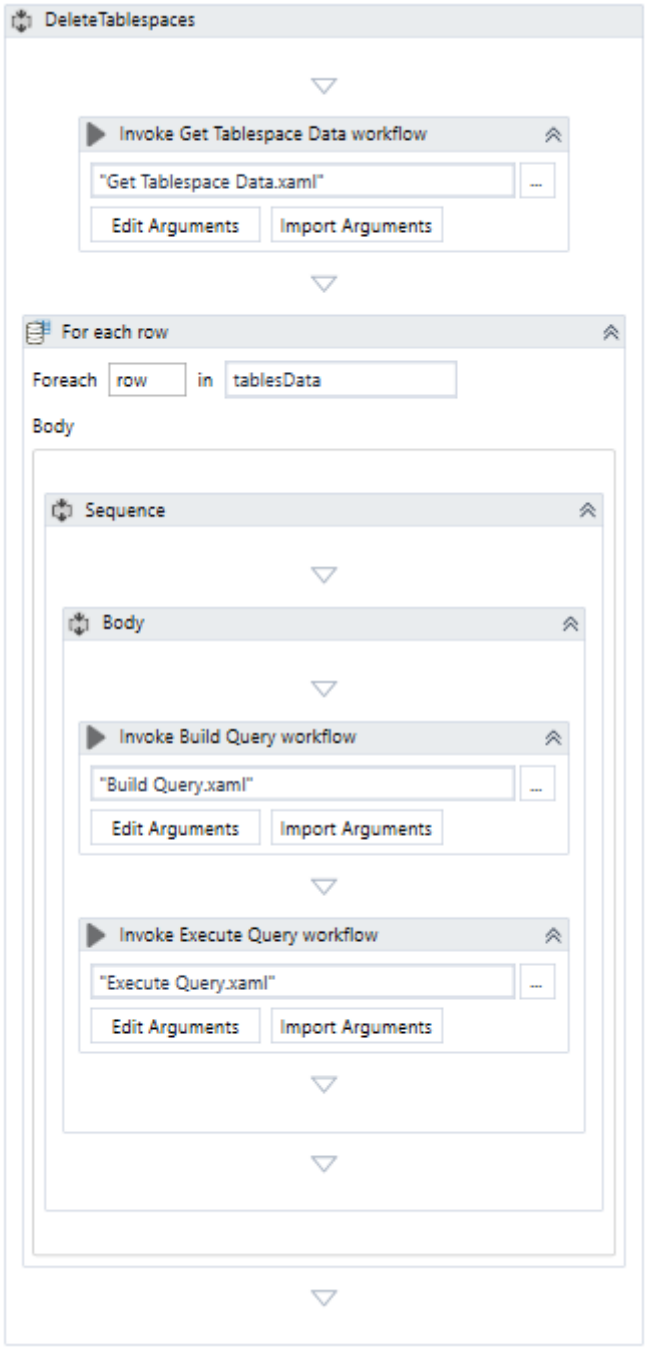


Figure 4-7: Diagram of the drop of tablespaces

This is the main file of the project, with the pseudocode and explanation of specific workflows:



Drop Tablespaces:

- tablesData= GetTablespacesData()
- For each row in tablesData:
  - query = BuildQuery(row)
  - ExecuteQuery(query)

Figure 4-8: Automation of drop tablespaces in UiPath

Get Tablespaces Data.xaml:

It is a workflow which reads the names of the tablespaces that are going to be dropped from an Excel file. It stores them in a data table (UiPath does not allow to read a range of an Excel file into an array), which is returned to the main workflow. It also adds that data

table to the Excel file “tablespace_existing.xlxs”, which keeps track of the existing tablespaces.

Build Query.xaml:

It is a workflow which receives the name tablespace that is going to be deleted. It builds the query to drop the tablespace, and returns it to the main workflow.

Execute Query.xaml:

It is a workflow which receives a string as an argument, and types that string in the SQL Worksheet of SQL Developer. After typing it, a shortcut for running the query is sent to the program.

4.2 Use Case 3: Freeing of locked resources

This is the diagram of the use case:

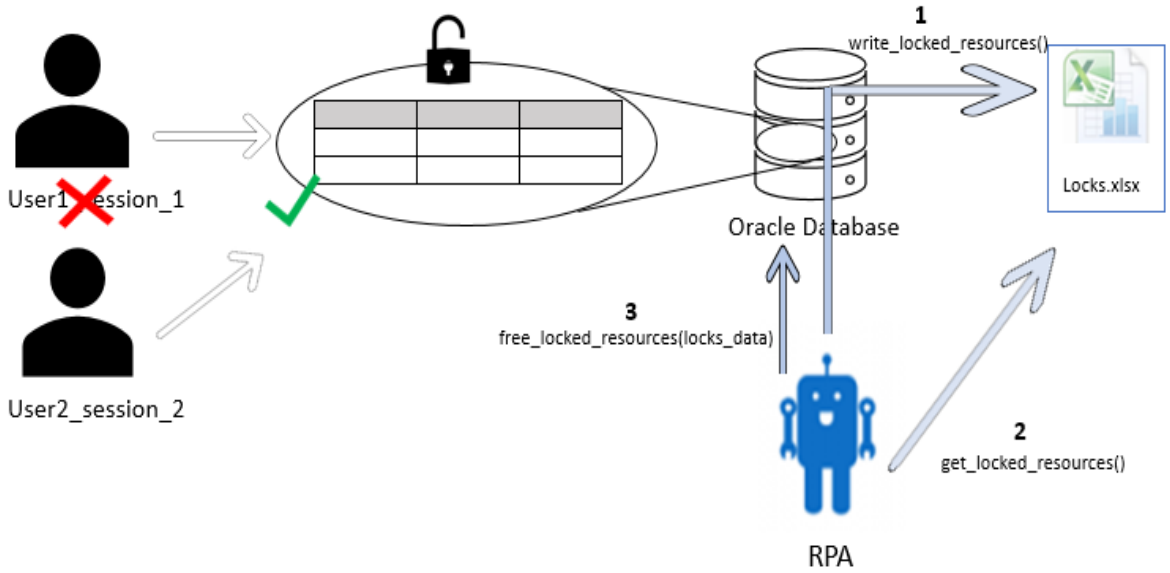
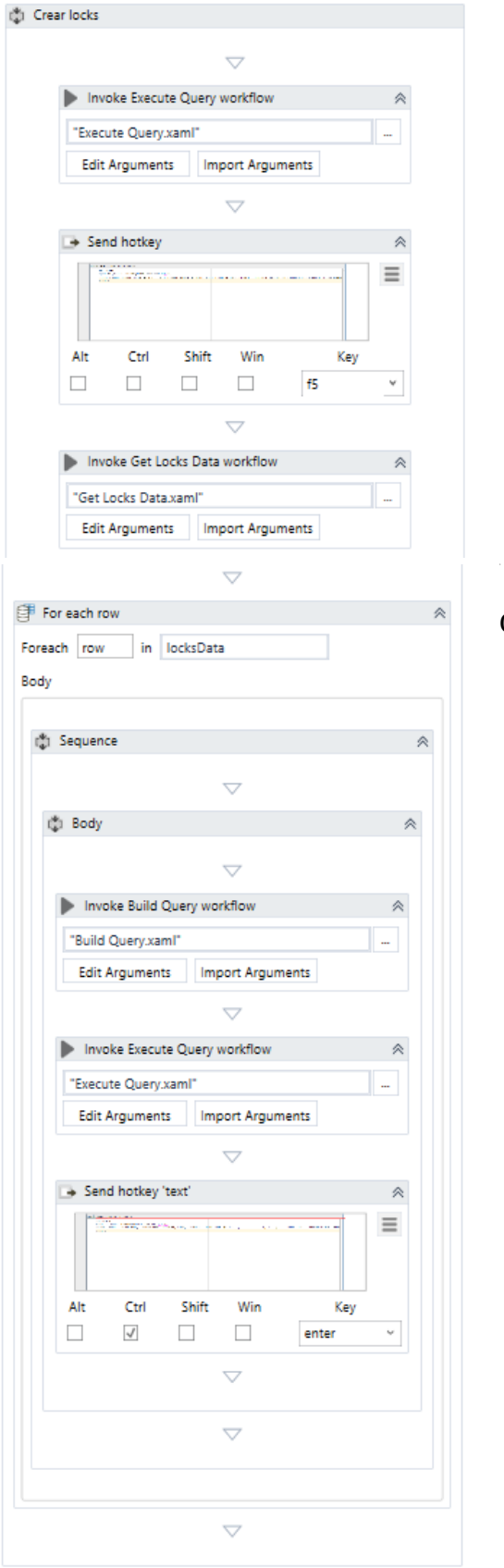


Figure 4-9: Diagram of the freeing of a locked table

This is the main file of the project, with the pseudocode and explanation of specific workflows:



Clear locks:

- ExecuteQuery
  (query_write_locked_resources)
- SendHotkey(f5)
- locksData = GetLocksData()
- For each row in LocksData:
  - query = BuildQuery(row)
  - ExecuteQuery(query_kill_session)
  - SendHotkey(Ctrl + enter)

Figure 4-10: Automation of freeing locked resources in UiPath

Execute Query.xaml:

In this case it is a little different. It is a workflow which receives a string as an argument, and types that string in the SQL Worksheet of SQL Developer. It does not execute the query like in the previous use cases. Queries will be executed in the "Send hotkey" activity.
The first query gets the inactive sessions locking resources and saves the result in an Excel file. The second query kills those inactive sessions one by one.

Send Hotkey activity:

It is an activity which sends the command received as argument. It is not the same argument both times because with "ctrl + enter" only one line from the SQL Worksheet is run, and with "f5" many lines are executed.

Get Locks Data.xaml:

It is a workflow which reads the information of the locked resources from an Excel file. It stores them in a data table, which is returned to the main workflow.

Build Query.xaml:

It is a workflow which receives the data of the locked resource and the locking session. It builds the query to kill the session, and returns it to the main workflow.

5 Tests and results

After the research, we have focused on automating three main use cases and it is what we will test now:

- Creation and drop of users.
- Creation and drop of tablespaces.
- Free of locked resources.

Now that we have automated them with a RPA, it is time to test our work and compared its efficiency with people´s performance. For this test, we solved the use cases in three different ways:

- Visually on SQL Developer. Since that program is done to make DBA tasks easier, it should not be the slowest. This will be performed by someone who knows the platform and knows how to solve the problems.
- "DBA mode". This way we will have an experienced DBA typing and executing the necessary queries on the SQL Worksheet. This can only be done by someone who perfectly knows SQL and Oracle Databases.
- By running a RPA. This can be done by anyone. He does not even need any technical knowledge of Oracle Databases.

We did this test with only an ideal performance. This means that no mistakes were made during the execution. Since humans are much more likely to make mistakes, this condition clearly benefits humans against a computer program, because we are more likely to make mistakes

The names and features were randomly generated for each of the tasks, so the conditions for all processes were the same.

Also, we did the test with two different amounts of work:

- With three repetitions: creation and drop of three users, creation and drop of three tablespaces, and unlock on three locked resources.
- With ten repetitions: creation and drop of ten users, creation and drop of ten tablespaces, and unlock on ten locked resources

The measured times, with 3 and 10 repetitions of each, to execute those tasks are shown in these two tables:

3 REPETITIONS	Visually	DBA	RPA
Create users	2:00	1:10	0:30
Drop users	0:25	0:45	0:30
Create tablespaces	2:40	4:00	1:20
Drop tablespaces	0:50	1:30	0:45
Free locked resources	-	2:20	0:15
TOTAL	5:45 + 2:20 = 8:05	9:45	3:20

Table 5-1: Performance test with three repetitions. Time is in MM:SS format

10 REPETITIONS	Visually	DBA	RPA
Create users	5:00	2:30	1:40
Drop users	1:10	1:40	1:30
Create tablespaces	8:00	11:30	4:40
Drop tablespaces	2:30	4:10	1:50
Free locked resources	-	4:10	0:30
TOTAL	16:40 + 4:10 = 20:50	24:00	10:10

Table 5-2: Performance test with ten repetitions. Time is in MM:SS format

After analyzing the results, we can arrive to a few conclusions:

- In shorts tasks, humans are equal, or even faster than a RPA.
- As shown in the last table, in longer and repetitive processes a RPA is two and even three times faster than a human.

This means that for people who know RPA and its strengths, RPA can save a lot of time. In the beginning it will seem difficult and long to automate processes, but it has been proved how it is worth it.

6 Conclusions and future work

6.1 Conclusions

The main goal of this work was to study RPA and its possible applications to help people with work. Specifically, we studied its appliance to the administration of Oracle Databases. We automated some tasks currently performed by DBAs and compared the work done by RPA with the one done by humans. We arrived to the following conclusions:

- RPA executes tasks based on the UI and interacts with it just like a person would do.
- RPA is much faster than a person, between two and three times in most of the cases that we studied. Also, RPA does not make mistakes.
- We need to establish and program the rules of the process which is going to be automated. After a RPA automation is developed, it can be executed automatically and independently. This allows people to focus on more complex tasks.
- They are really effective with repetitive and rule-based processes. But RPA is not prepared yet for working on tasks that they were not taught to solve. They have a lack on intuition and creativity.

6.2 Future work

Due to the extent of this work, it was not possible to study everything wanted. But some extensions which could be studied and added in the future are:

- Add error control. The developed automations work perfectly with cases when everything goes as expected, but it would be helpful controlling unexpected situations and how to act when they happen. Some examples for those undesired situations would be: losing the connection with the Oracle Database, or having UI changes.
- Design of a possible implementation to run those automated processes on virtual environments on the cloud. This way, RPA could be externalized and executed automatically. It would be especially useful for companies with many RPA automations.

Study of the technology called Intelligent Process Automation (IAP), which “is an emerging set of new technologies that combines fundamental process redesign with robotic process automation and machine learning. It is a suite of business-process improvements and next-generation tools” [27].

References

[1] "Robotic Process Automation & Artificial Intelligence", https://hackernoon.com/robotic-process-automation-artificial-intelligence-b136e9a2f91c
[2] "What is RPA? A Guide to Robotic Process Automation", http://blog.symphonyhq.com/what-is-rpa
[3] "A Q&A on RPA", https://www.uipath.com/blog/a-qa-on-rpa
[4] UiPath, https://www.uipath.com/
[5] "Introduction to Oracle Database", https://docs.oracle.com/en/database/oracle/oracle-database/12.2/cncpt/introduction-to-oracle-database.html#GUID-A42A6EF0-20F8-4F4B-AFF7-09C100AE581E
[6] "Process architecture", https://docs.oracle.com/en/database/oracle/oracle-database/12.2/cncpt/process-architecture.html#GUID-85D9852E-5BF1-4AC0-9E5A-49F0570DBD7A
[7] "Physical storage structures", https://docs.oracle.com/en/database/oracle/oracle-database/12.2/cncpt/physical-storage-structures.html#GUID-FFA872E1-7F63-4DC5-8A35-F21394AB4595
[8] "Logical storage structures", https://docs.oracle.com/en/database/oracle/oracle-database/12.2/cncpt/logical-storage-structures.html#GUID-13CE5EDA-8C66-4CA0-87B5-4069215A368D
[9] "Introduction to Tablespaces, Datafiles, and Control Files", https://docs.oracle.com/cd/B19306_01/server.102/b14220/physical.htm#i2009
[10] "Administering User Privileges, Roles, and Profiles", https://docs.oracle.com/cd/B19306_01/network.102/b14266/admusers.htm#DBSEG10000
[11] "Data Concurrency and Consistency", https://docs.oracle.com/cd/B19306_01/server.102/b14220/consist.htm#i13945
[12] Common Oracle DBA Tasks, https://docs.oracle.com/cd/B16276_01/doc/server.102/b14196/intro003.htm
[13] "DBA Common Task Descriptions", https://wpollock.com/Oracle/DBA-Tasks.htm
[14] "Robotic Process Automation (RPA) an introduction", https://www.cortex.co.uk/news/rpa-an-introduction/
[15] "Robotic Process Automation: Why It Can´t Replace Humans", https://www.appway.com/screen/blog/id/Robotic-Process-Automation-Why-It-Can-t-Replace-Humans-1519763602473
[16] "The Big 3: Leading RPA Vendors", https://www.nimbulconsulting.com/rpa-automatizacion-robotica-de-procesos/
[17] "The Big 3: Leading RPA Vendors", http://3qilabs.com/the-big-3-leading-rpa-vendors/
[18] " UiPath's RPA Enterprise Platform", https://www.uipath.com/platform
[19] "About the Core Activities Pack", https://activities.uipath.com/docs/about-the-core-activities-pack
[20] "Introduction", https://studio.uipath.com/docs/introduction
[21] "About Selectors", https://studio.uipath.com/docs/about-selectors
[22] "About Data Scrapping", https://studio.uipath.com/docs/about-data-scraping
[23] "About image and text automation", https://studio.uipath.com/docs/about-image-and-text-automation
[24] "Oracle locks", https://dbpost.wordpress.com/tag/vlocked_object/
[25] "How are tables unlocked in Oracle", https://www.quora.com/How-are-tables-unlocked-in-Oracle
[26] "Intelligent process automation: The engine at the core of the next-generation operating model", https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/intelligent-process-automation-the-engine-at-the-core-of-the-next-generation-operating-model

Glossary

API	Application Programming Interface.
Data scrapping	Technique to extract data from another program.
DBA	Database Administrator.
DBMS	Database Management System
IPA	Intelligent Process Automation
OCR	Optical Character Recognition
Oracle Database	Database Management System from Oracle
RPA	Robotic Process Automation
Selector	Element used to store the attributes of an element from the user interface
SQL	Structured Query Language
UI	User Interface